

**Ejercicio 1:** Dada la siguiente expresión en una cadena de texto: “2+2” ¿cómo podría evaluarse usando una función predefinida del lenguaje Javascript para, de esa forma obtener el resultado numérico: 4? Escribe la sentencia.

```
let text = "2 + 2";

let result = eval(text);
```

**Ejercicio 2:** Crea una función que reciba como parámetro una cadena y devuelva el número de vocales que contiene. Seguidamente invoca a esa función con el valor de entrada: “Entorno Cliente 4”, de modo que al cargar la página, muestre por la consola el resultado correcto: 6. Por último convierte la función anterior en una función anónima (y comprueba que siga funcionando correctamente)

```
function nVocales(){

    var cadena='Entorno Cliente 4';

    var caracter;

    var vocales=0;

    for(var i=0;i<cadena.length-1;i++){

        caracter= cadena.charAt(i).toLowerCase();

        if(caracter=='a' || caracter=='e' || caracter.toLowerCase()=='i' ||
caracter.toLowerCase()=='o' || caracter.toLowerCase()=='u' ){

            vocales++;

            console.log(vocales);

        }else{

            break;

        }

    }

    document.write('El numero de vocales es : '+ vocales);

}
```

**Ejercicio 3: Crea una función que reciba un número n y devuelva un array relleno con los n primeros números primos.**

**Imprime el array relleno en por la página web (por ejemplo en un div) separando cada número por el carácter punto y coma (;).**

```
var nprimos=prompt('¿Cuántos primos quieres imprimir?')
```

```
function primo(nprimos) {  
  var count = 0;  
  for (var i = 1; i < nprimos; i++) {  
    if (i % 2 == 0 || i % 3 == 0 || i % 5 == 0) {  
      count++;  
    }  
  }  
  if (count == 2) {  
    document.write(i + " ");  
  }  
  count = 0;  
}
```

**Ejercicio 4: Declara un objeto que represente a una persona. Para ello incluye los atributos: nombre, apellidos y edad, y el método toString que devuelva una cadena de esta forma: “apellidos, nombre – edad años”. Ejemplo: Ruiz de Lopera, Manuel – 75 años**

```
function Persona(nombre, apellido, edad) {  
  this.nombre = nombre;  
  this.apellido=apellido  
  this.edad = edad;  
  this.toString= apellido+', ' + nombre + '-'+ edad;  
}  
  
var persona1=new Persona('Manuel', 'Ruiz de Lopera', 75);
```

```
document.write(persona1.toString);
```

**Ejercicio 5: Añade un método al prototipo del objeto Persona del ejercicio anterior que muestre una alerta saludando de la siguiente manera: ¡Hola, *nombre*!**

```
function Persona(nombre, apellido, edad) {  
  
    this.nombre = nombre;  
  
    this.apellido = apellido;  
  
    this.edad = edad;  
  
    this.toString = apellido + ", " + nombre + "-" + edad;  
  
    persona.prototype.saludar = function () {  
  
        alert("Hola " + this.name.nombre);  
  
    };  
  
}  
  
var persona1 = new Persona("Manuel", "Ruiz de Lopera", 75);  
  
persona1.prototype.saludar();
```

**Ejercicio 6: Define una función constructora para crear objetos de tipo persona con los mismo atributos**

**y métodos que la del ejercicio 4.**

**Instancia 4 personas diferentes e Introdúcelas en un array. A continuación ordena el array**

**primero por apellido, luego por nombre y finalmente por edad. Utiliza para ello la función de**

**los arrays vista en clase.**

```
if (  
  
    a.completoNombre.apellido.toLowerCase() >  
  
    b.completoNombre.apellido.toLowerCase()  
  
) {  
  
    x = 1;  
  
} else if (  
  

```

```

a.completoNombre.apellido.toLowerCase() <
b.completoNombre.apellido.toLowerCase()
) {
    x = -1;
} else {
    if (
        a.completoNombre.nombre.toLowerCase() >
        b.completoNombre.nombre.toLowerCase()
    ) {
        x = 1;
    } else if (
        a.completoNombre.nombre.toLowerCase() <
        b.completoNombre.nombre.toLowerCase()
    ) {
        x = -1;
    } else {
        if (a.edad > b.edad) {
            x = 1;
        } else {
            x = -1;
        }
    }
    return x;
}
});

```

**Ejercicio 7: Rellena una array de 10 posiciones con n números enteros aleatorios entre 0 y 10. Después realiza una función que reciba el array y devuelva un objeto con el mayor y el menor número del array.**

```
function generarRandom() {  
    return Math.floor(Math.floor(Math.random() * (11 - 1) + 1));  
}  
  
const numeros = [];  
  
for (var i = 0; i <= 9; i++) {  
    numeros.push(generarRandom());  
}  
  
function verMax(array) {  
    this.maximo = Math.max(...numeros);  
    return this.maximo;  
}  
  
function verMin(array) {  
    this.min = Math.min(...numeros);  
    return min;  
}
```

**Ejercicio 8: Dado un array numérico, definir una función que determine si es simétrico o no. Esto es, si los números que están a la misma distancia del centro del array son iguales. Ejemplos:**

**Ejercicio 9: Para un array de personas ordenado como en el ejercicio 6, muéstralos en la página html como una lista desordenada (de viñetas) en orden inverso a como estaba ordenado el array,**

pero sin usar para ello la propiedad `length` ni ningún número mayor que 1.

```
var concatreves = "";

for (let i = 0; i < arraypersonas.length; i++) {

    var mostrarAlreves =ordenado.reverse();

    concatreves =concatreves +

        mostrarAlreves.completoNombre.apellido+' '
+mostrarAlreves.completoNombre.nombre+' ' +mostrarAlreves.edad +

        "<br>";

}
```

### Ejercicio 10;

```
console.log('parent: ' + parent.get());
```

Imprime 80 ya que es el valor que se le a asignado al crearlo.

```
console.log('child: ' + child.get());
```

Imprime 40 ya que al heredar lo hace con 80 pero le actualizamos el dato con `"child.val=40;"`

```
console.log('grandchild: ' + grandchild.get());
```

Imprime 80 ya que hereda de parent el atributo y su valor y en ningun punto se actualiza.